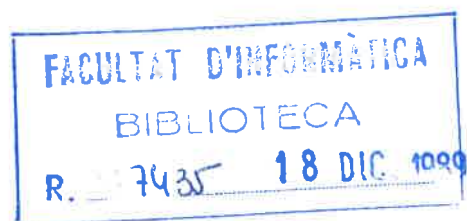


• 1400008470
còpia 1

El algoritmo de completión de Knuth-Bendix

M.P. Nivela
R. Peña

Report LSI-88-25



Abstract: An introduction to the confluence and finite termination properties of rewriting systems is presented and also to the Kunth-Bendix completion algorithm is presented.

Resum: Es presenta una introducció a les propietats de confluència i terminació finita dels sistemes de reescriptura de termes i a l'algorisme de completió de Kunth-Bendix.

El algoritmo de completión de Knuth-Bendix

M^a Pilar Nivelá Alós

Ricardo Peña Mari

Indice

- 1. Planteamiento del problema**
- 2. Resumen de conceptos previos**
- 3. Confluencia**
- 4. Decidibilidad de la confluencia**
- 5. El algoritmo de completión**

1. Planteamiento del problema

Una especificación algebraica ecuacional SP define una teoría satisfecha por diferentes álgebras que constituyen la clase Alg_{SP} . Se han introducido las reglas del cálculo ecuacional que permiten deducir, a partir de SP, teoremas de la forma $t_1 = t_2$, $t_1, t_2 \in T_{\Sigma}(X)$, válidos en toda la clase Alg_{SP} .

El cálculo ecuacional es *completo y correcto*, es decir, permite deducir cualquier teorema válido en Alg_{SP} y, además, todos los teoremas que se deducen mediante este cálculo ecuacional son válidos en Alg_{SP} . Pero no es un procedimiento *efectivo* para decidir la validez de una ecuación dada. La única esperanza de lograrlo consistiría en enumerar todas las ecuaciones deducibles de SP (normalmente infinitas) y ver si alcanzamos o no la ecuación propuesta. Este procedimiento puede llegar (si se le deja suficiente tiempo) a validar la ecuación pero no a refutarla.

La **reescritura**, consistente en considerar las ecuaciones E de SP orientadas en uno de los dos sentidos posibles y en aplicarlas sólo en el sentido determinado, puede suministrar dicho procedimiento efectivo bajo ciertas condiciones. Si llamamos R al conjunto de las ecuaciones de E orientadas en algún sentido, entonces indicaremos por \rightarrow_{R} la aplicación de una de las reglas de R, por \rightarrow_{R}^+ la aplicación de una o más reglas de R, por \rightarrow_{R}^* la aplicación de cero o más reglas de R y por $\leftrightarrow_{\text{R}}^*$ la aplicación de cero o más veces las reglas de R en ambos sentidos. Se cumple entonces que $=_{\text{E}}$ coincide con $\leftrightarrow_{\text{R}}^*$.

Un primer objetivo de la reescritura es el de *evaluar* términos, es decir, simplificar un término reescribiéndolo cuantas veces sea posible. Para que esto tenga sentido deberá ocurrir que el número total de reescrituras a aplicar sea finito. Es la propiedad de **terminación finita** o **noetherianidad**, que garantiza que todo término $t \in T_{\Sigma}(X)$ es reescrito un número finito de veces hasta alcanzar un término t' que no es posible reescribir más. Se dice entonces que t' es una **forma normal** de t y se indica por $t' = t \downarrow$.

Un segundo objetivo de la reescritura es proporcionar un método más sencillo para decidir si $t \leftrightarrow_{\text{R}}^* t'$ (o lo que es igual, $t =_{\text{E}} t'$) substituyendo este cálculo por otro más sencillo consistente en calcular la forma normal de t , $t \downarrow$, la forma normal de t' , $t' \downarrow$, y comprobando si son iguales. Para que este segundo cálculo sea equivalente al primero es necesario que se cumpla una segunda propiedad, llamada **confluencia**. Esta asegura que el resultado de reescribir t no depende de las reglas que se escojan durante el proceso de reescritura, pues en cada paso podrían ser aplicables varias reglas, o la misma regla, a diferentes subterminos de t .

Un sistema de reescritura que posea ambas propiedades se dice **canónico**. En esta situación, todo término t tiene forma normal única y el procedimiento de decidir si $t_1=t_2$ es válida en Alg_{SP} es efectivo. Consiste simplemente en reescribir t_1 y t_2 y comparar sus respectivas formas normales. El teorema es válido si y solo si éstas coinciden.

Ambas propiedades son, en general, indecidibles. Existen sin embargo condiciones suficientes, comprobables algorítmicamente, que garantizan la noetherianidad de un conjunto de reglas. En cuanto a la confluencia, es decidible si el sistema ya es noetheriano. Mejor aún, un sistema de reglas noetheriano no confluente puede, en muchos casos, ser transformado en un sistema confluente equivalente. El procedimiento para hacerlo es el llamado **algoritmo de completación**. En realidad no es un algoritmo sino un semi-algoritmo. Si termina sin fallo llegará a conseguir un sistema canónico a partir del sistema inicial. Si termina con fallo o no termina, no es posible afirmar si existe o no dicho sistema canónico equivalente. El "algoritmo" puede partir también de un conjunto de ecuaciones no orientadas y, si termina, suministrar un sistema canónico de reglas que especifica la misma teoría algebraica que las ecuaciones de partida. Usado de este modo, se convierte en algo similar a un compilador que, dada una especificación ecuacional, la "compila" y obtiene el "codigo ejecutable" equivalente en forma de sistema de reescritura canónico.

2. Resumen de conceptos previos

Un término t de $T_{\Sigma}(X)$ puede interpretarse como una aplicación parcial de N^{+*} (monoide libre sobre N^+) en $\Sigma \cup X$

$$t : N^{+*} \rightarrow \Sigma \cup X$$

cuyo dominio, que denotaremos por $\text{Pos}(t)$ y que recibe el nombre de **conjunto de posiciones de t** , verifica

- a) la palabra vacía λ pertenece a $\text{Pos}(t)$
- b) $ui \in \text{Pos}(t)$ sii $u \in \text{Pos}(t)$ y $1 \leq i \leq \text{aridad}(t(u))$

Si se indica por \leq el orden parcial prefijo sobre N^{+*} , el conjunto $\text{Pos}(t)$ es cerrado para este orden.

2. Se indica por t_u el subtérmino de t en la posición u y que viene definido por

- a) $\text{Pos}(t_u) = \{ v \in \mathbf{N}^{+*} \mid uv \in \text{Pos}(t) \}$
 b) $t_u(v) = t(uv)$

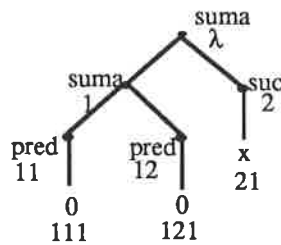
Si t' es un subtérmino de t , escribiremos $t' \ll t$.

3. Se indica por $t[u \leftarrow t']$ el término que se obtiene a partir de t al sustituir el subtérmino t_u por t' y que viene definido por:

- a) $\text{Pos}(t[u \leftarrow t']) = \{ v \in \mathbf{N}^{+*} \mid v \in \text{Pos}(t) \text{ y } \text{no}(u \leq v) \} \cup \{ v \in \mathbf{N}^{+*} \mid v = uv' \text{ y } v' \in \text{Pos}(t') \}$
 b) $t[u \leftarrow t'](uv') = t'(v')$ si $v' \in \text{Pos}(t')$
 c) $t[u \leftarrow t'](v) = t(v)$ si $v \notin \text{Pos}(t')$

Por ejemplo, para el término $t = \text{suma}(\text{suma}(\text{pred}(0), \text{pred}(0)), \text{suc}(x))$ se tiene que

$$\text{Pos}(t) = \{ \lambda, 1, 2, 11, 12, 21, 111, 121 \}$$



$$\text{y } t(\lambda) = t(1) = \text{suma}, t(2) = \text{suc}, t(11) = t(12) = \text{pred}, t(111) = t(121) = 0, t(21) = x.$$

Definición

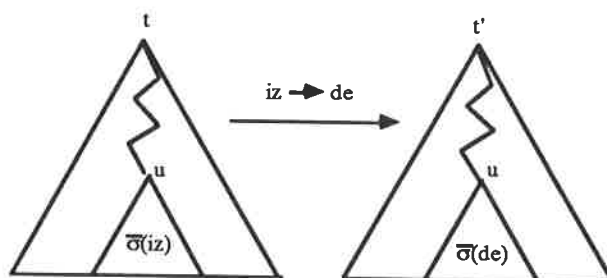
Una **regla de reescritura**, escrita de la forma $iz \rightarrow de$, es un par de términos (iz , de) con $iz, de \in T_{\Sigma}(X)$, tales que $\text{var}(de) \subseteq \text{var}(iz)$.

Un **sistema de reescritura** R es un conjunto de reglas de reescritura.

Definición

Sea R un sistema de reescritura. La **relación de reducción** \rightarrow_R es la relación más fina sobre $T_{\Sigma}(X)$ que contiene a R y es estable por sustitución y reemplazo. Estable por sustitución quiere decir que si $t \rightarrow_R t'$ entonces $\bar{\sigma}(t) \rightarrow_R \bar{\sigma}(t')$, para toda sustitución $\sigma: \text{var}(t) \rightarrow T_{\Sigma}(X)$. Estable por reemplazo significa que si $t_1 \rightarrow_R t'_1$ entonces $t[u \leftarrow t_1] \rightarrow_R t[u \leftarrow t'_1]$ para toda posición $u \in \text{Pos}(t)$.

Esta definición es equivalente a decir que $t \rightarrow_R t'$ (t se reescribe en t') si existe una regla $iz \rightarrow de$ en R , una sustitución $\sigma: \text{var}(iz) \rightarrow T_\Sigma(X)$ y una posición $u \in \text{Pos}(t)$ tales que $t|_u = \overline{\sigma}(iz)$ y $t' = t[u \leftarrow \overline{\sigma}(de)]$.



Denotaremos por \rightarrow_R^+ la clausura transitiva de \rightarrow_R , por \rightarrow_R^* la clausura reflexiva y transitiva de \rightarrow_R y por \leftrightarrow_R^* la clausura reflexiva, simétrica y transitiva de \rightarrow_R .

Si no hay ambigüedad respecto a quién es R , se omitirá en las anteriores denotaciones, escribiendo simplemente \rightarrow , \rightarrow^+ , \rightarrow^* y \leftrightarrow^* .

Resultado

Dada una especificación $SP = (S, \Sigma, E)$ y un sistema de reescritura R obtenido orientando las ecuaciones de E en alguno de los dos sentidos posibles, se tiene:

Para todo $t, t' \in T_\Sigma$, $t \leftrightarrow_R^* t'$ si y solo si $t \equiv_E t'$

Más aún, para todo $t, t' \in T_\Sigma(X)$, $t \leftrightarrow_R^* t'$ si y solo si $SP \vdash t = t'$

Definición

Un término $t \in T_\Sigma(X)$ es **reducible** según R si existe t' tal que $t \rightarrow t'$. En caso contrario t se dice **irreducible**.

Una **forma normal** de t , si existe, es un término irreducible t' , tal que $t \rightarrow^* t'$.

Definición

Un sistema de reescritura R es **noetheriano** o de **terminación finita**, si no existe una cadena infinita de reducciones $t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n \rightarrow \dots$.

El problema de determinar si un sistema de reescritura es noetheriano es, en general, indecidible. No obstante, es posible encontrar algorítmicamente en muchos casos condiciones suficientes que garantizan la noetherianidad. Estos algoritmos tratan de demostrar la existencia en $T_\Sigma(X)$ de un **orden de reducción**.

Definición

Dado R , un **orden de reducción** en $T_{\Sigma}(X)$ es un orden parcial estricto $>$ *bien fundado* (sin cadenas decrecientes infinitas) compatible con la reescritura, es decir, si $t \rightarrow_R t'$ entonces $t > t'$.

Definición

Un orden parcial estricto $>$ es **monótono** si para todo t, t' y para todo $f \in \Sigma$: $t > t'$ implica $f(\dots t \dots) > f(\dots t' \dots)$.

Resultado

Un sistema de reescritura R es noetheriano si y solo si existe un orden bien fundado monótono $>$ tal que, para toda regla $iz \rightarrow de \in R$ se cumple $iz > de$.

Se resumen ahora los conceptos de *instancia* de un término, *unificación* y *superposición* de términos, que serán utilizados en los apartados siguientes.

Definición

Diremos que t' se **sumerge en** t , y se indica por $t \leq t'$, si existe una sustitución $\sigma: \text{var}(t) \rightarrow T_{\Sigma}(X)$, $\text{var}(t) \subseteq X$, tal que $t' = \bar{\sigma}(t)$. Se dice también que t' es una **instancia** de t . La sustitución σ se denomina **instanciación** de t a t' .

Definición

Diremos que σ' se **sumerge en** σ sobre X , y se indica por $\sigma \leq \sigma' [X]$, si existe ρ tal que para todo $x \in X$, $\bar{\rho} \circ \bar{\sigma}(x) = \sigma'(x)$.

Ejemplo: $\sigma'(x) = f(y, y)$ está sumergida en $\sigma(x) = f(z, y)$ sobre $X = \{x\}$

Resultados

Si t' se sumerge en t , la instanciación es única. Es decir, si $t' = \bar{\sigma}(t)$ y $t' = \bar{\sigma}'(t)$, $\sigma, \sigma': \text{var}(t) \rightarrow T_{\Sigma}(X)$, entonces $\sigma = \sigma'[\text{var}(t)]$ es decir, σ y σ' coinciden sobre el dominio $\text{var}(t)$.

La relación de inmersión \leq en $T_{\Sigma}(X)$ es un preorden bien fundado llamado **preorden de inmersión**.

Si $\sigma \leq \sigma' [X]$ entonces, para todo $t \in T_{\Sigma}(X)$, $\bar{\sigma}(t) \leq \bar{\sigma}'(t)$.

Definición

Unificar dos términos t y t' de $T_{\Sigma}(X)$ es resolver la ecuación $t = t'$ en $T_{\Sigma}(X)$. Se dice que $\sigma: \text{var}(t) \cup \text{var}(t') \rightarrow T_{\Sigma}(X)$ es un **unificador** de t y t' si $\bar{\sigma}(t) = \bar{\sigma}(t')$. Si existe algun unificador de t y t' , se dice que ambos términos son **unificables**.

Se llama **unificador más general** de t y t' , y se indica $umg(t, t')$, al menor unificador sobre $var(t) \cup var(t')$ según el preorden \leq .

Resultado

Si t y t' son unificables, el unificador más general $umg(t, t')$ es único.

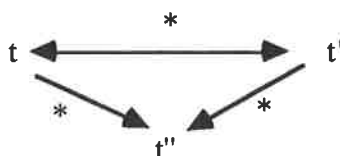
Definición

Diremos que el término t' se **superpone** al término t en la posición $u \in Pos(t)$ con sustitución σ , si $\sigma = umg(t', t_u)$.

3. Confluencia

Definición

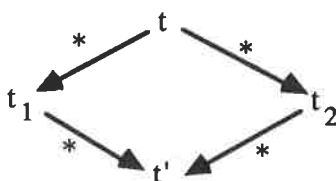
Un sistema de reescritura R es **Church-Rosser** si para todo par de términos $t, t' \in T_{\Sigma}(X)$ tales que $t \leftrightarrow^* t'$, existe $t'' \in T_{\Sigma}(X)$ tal que $t \rightarrow^* t''$ y $t' \rightarrow^* t''$.



La propiedad Church-Rosser permite reducir el problema de estudiar la congruencia engendrada en $T_{\Sigma}(X)$ por un conjunto de ecuaciones E , al problema más sencillo de reescribir términos según un conjunto de reglas R obtenido orientando las ecuaciones de E . Si esta reescritura termina, el procedimiento de decisión para comprobar $t \equiv_E t'$ (equivalente, como se ha dicho, a $t \leftrightarrow_R^* t'$) es efectivo. Una propiedad demostrada equivalente a la de Church-Rosser es la de **confluencia**.

Definición

Un sistema de reescritura R es **confluente** si para todo $t, t_1, t_2 \in T_{\Sigma}(X)$ tales que $t \rightarrow^* t_1$ y $t \rightarrow^* t_2$, existe $t' \in T_{\Sigma}(X)$ tal que $t_1 \rightarrow^* t'$ y $t_2 \rightarrow^* t'$.



Esta propiedad garantiza que, independientemente de qué reglas apliquemos a un término (si varias son aplicables) y a qué subtérminos se apliquen (si varios son reducibles por la misma o diferentes reglas), el resultado final, si existe, será único. Ello permite reducir términos sin necesidad de "vuelta atrás" para deshacer la aplicación de una regla previa, caso frecuente de encontrar en otros sistemas basados en reglas. La propiedad de confluencia también expresa el carácter "funcional" de las operaciones de un álgebra: la evaluación de un término en un álgebra a partir de unos valores iniciales da como resultado un valor único ya que todas las operaciones involucradas son deterministas.

Definición

Un sistema de reescritura R confluente y noetheriano, se dice **canónico**.

Resultados

Un sistema de reescritura R es Church-Rosser si y solo si es confluente.

Si un sistema de reescritura R es noetheriano, todo término t admite al menos una forma normal.

Si un sistema R es confluente, todo término t tiene como máximo una forma normal (puede no tener ninguna).

Si un sistema R es canónico, todo término t posee una **única** forma normal. La denotaremos $t\downarrow$. Más aún, si $SP = (S, \Sigma, E)$ donde E son las reglas de R consideradas como ecuaciones, se tiene:

Para todo $t, t' \in T_\Sigma$, $t \equiv_E t'$ si y solo si $t\downarrow = t'\downarrow$

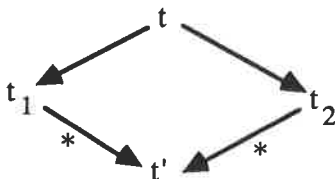
Para todo $t, t' \in T_\Sigma(X)$, $SP \vdash t = t'$ si y solo si $t\downarrow = t'\downarrow$

4. Decidibilidad de la confluencia

La comprobación de la confluencia de un sistema de reescritura es también un problema en general indecidible pero, para los sistemas de terminación finita, la propiedad se convierte en decidible. Ello es así porque, en esas condiciones, la confluencia se demuestra equivalente a otra propiedad más débil llamada *confluencia local*, a su vez equivalente a otra propiedad (convergencia de los *pares críticos*) comprobable algorítmicamente estudiando las posibles superposiciones de pares de reglas, siempre bajo condiciones de terminación finita. En esta sección se desarrollan los resultados relevantes que permiten decidir la confluencia de un sistema noetheriano de reescritura.

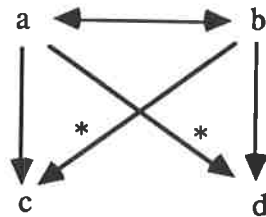
Definición

Un sistema de reescritura R es **localmente confluente** si para todo $t, t_1, t_2 \in T_\Sigma(X)$ tales que $t \rightarrow t_1$ y $t \rightarrow t_2$, existe $t' \in T_\Sigma(X)$ tal que $t_1 \rightarrow^* t'$ y $t_2 \rightarrow^* t'$.

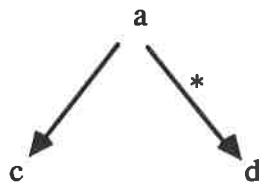


Obviamente la confluencia implica la confluencia local pero no a la inversa. Podemos

considerar, por ejemplo, el sistema de reescritura $R = \{ a \rightarrow b, b \rightarrow a, a \rightarrow c, b \rightarrow d \}$ es localmente confluente como puede verse observando el diagrama



pero no es confluente, pues el diagrama



no puede cerrarse.

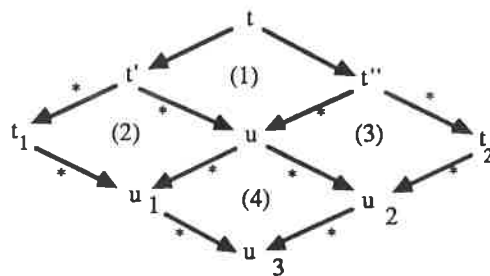
El siguiente teorema establece la equivalencia entre confluencia y confluencia local para sistemas de terminación finita.

Teorema de Newman

Sea R un sistema de reescritura noetheriano. R es confluente si y solo si R es localmente confluente.

Demostración

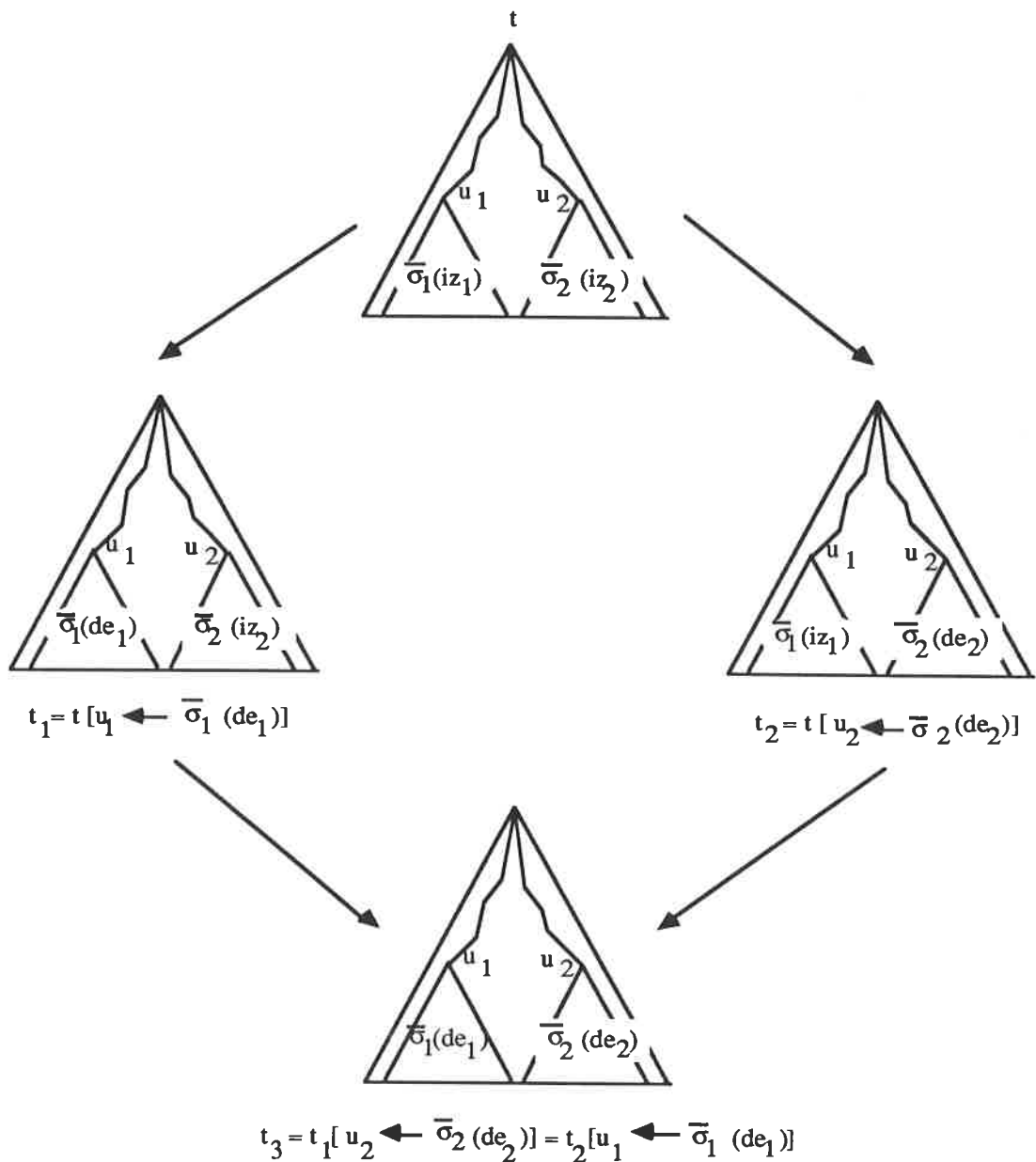
Se hará por inducción noetheriana sobre el número de pasos de reescritura. Sean $t, t_1, t_2 \in T_{\Sigma}(X)$ tales que $t \rightarrow^* t_1$ en n_1 pasos de reescritura y $t \rightarrow^* t_2$ en n_2 pasos de reescritura. Si $n_1 = 0$ ó $n_2 = 0$ R es trivialmente confluente para t . Supongamos $n_1 > 0$ y $n_2 > 0$. Entonces existen t', t'' tales que $t \rightarrow t', t \rightarrow t''$ y, por ser R localmente confluente, existe u que cierra el diagrama (1) de la figura:



Por hipótesis de inducción existen u_1, u_2 y u_3 que cierran los diagramas (2), (3) y (4) respectivamente. Luego R es confluente. ♦

Estudiaremos a continuación cómo se puede decidir la confluencia local de un sistema de escritura de terminación finita. Al tratar de reducir un término, se pueden dar diversas situaciones: supongamos que t se puede reducir utilizando dos reglas diferentes, $iz_1 \rightarrow de_1$ en la posición u_1 de t e $iz_2 \rightarrow de_2$ en la posición u_2 de t .

caso 1) u_1 y u_2 son disjuntas, es decir, ninguna posición es un prefijo de la otra.



En este caso siempre se tiene confluencia ya que será posible aplicar $iz_2 \rightarrow de_2$ a t_1 en la posición u_2 , e $iz_1 \rightarrow de_1$ a t_2 en la posición u_1 , de manera que t_1 y t_2 se reescribirán en el mismo término $t_3 = t_1 [u_2 \leftarrow \bar{\sigma}_2(de_2)] = t_2 [u_1 \leftarrow \bar{\sigma}_1(de_1)]$

Ejemplo: $R = \{ \text{pred}(\text{suc}(x)) \rightarrow x, \text{suma}(\text{pred}(x), y) \rightarrow \text{pred}(\text{suma}(x, y)) \}$

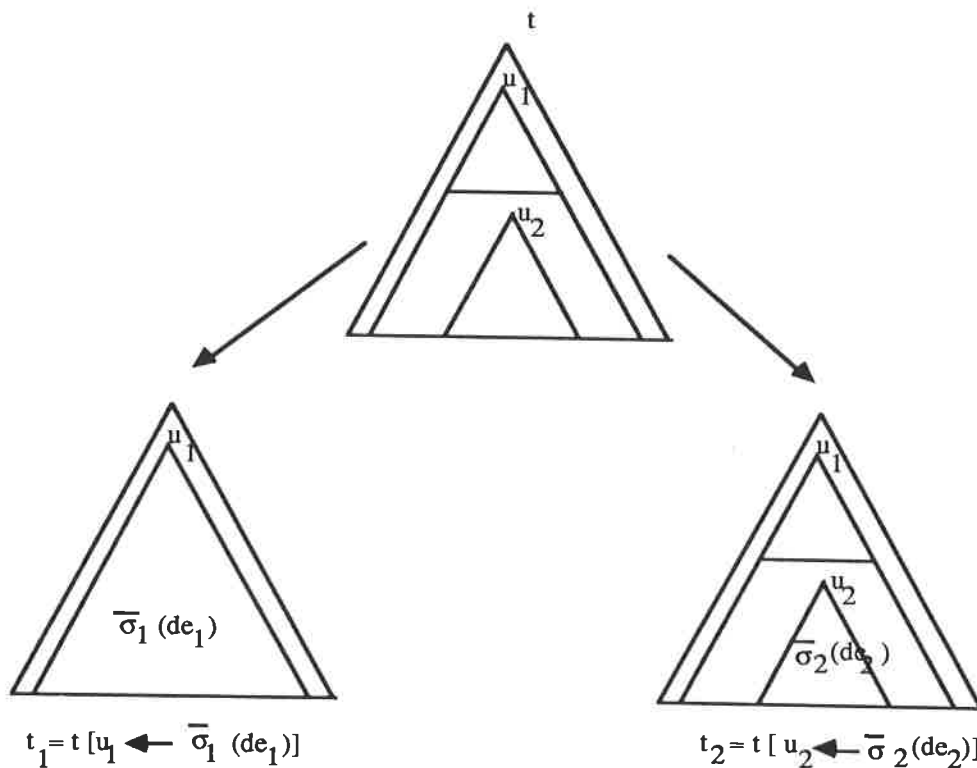
$$t = \text{suma} (\text{suma} (\text{pred} (x), y), \text{pred} (\text{suc} (x)))$$

$$t_1 = \text{suma} (\text{suma} (\text{pred} (x), y), x)$$

$$t_2 = \text{suma} (\text{pred} (\text{suma} (x, y)), \text{pred} (\text{suc} (x)))$$

$$t_3 = \text{suma} (\text{pred} (\text{suma} (x, y)), x)$$

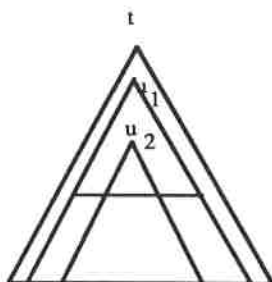
caso 2) la posición u_1 es prefijo de u_2 pero u_2 no "pertenece al patrón" definido por iz_1 . Más precisamente, u_2 no corresponde a una posición en la que iz_1 tiene un símbolo de operación. Este caso puede representarse gráficamente de la forma:



En este caso, t_2 sigue manteniendo intacto el "patrón" de iz_1 y, por tanto, siempre se puede aplicar esta regla a t_2 en la posición u_1 obteniendo t_3 . Si t_{u_2} sigue siendo subtérmino de $\bar{\sigma}_1(de_1)$ entonces t_1 se puede reducir aplicando $iz_2 \rightarrow de_2$ a dicho subtérmino, obteniéndose también t_3 . En caso contrario, t_1 es directamente t_3 .

Ejemplo: $R = \{ \text{pred}(\text{suc}(x)) \rightarrow x, \text{suma}(\text{pred}(x), y) \rightarrow \text{pred}(\text{suma}(x, y)) \}$
 $t = \text{suma}(\text{pred}(x), \text{suc}(\text{pred}(\text{suc}(y))))$
 $t_1 = \text{pred}(\text{suma}(x, \text{suc}(\text{pred}(\text{suc}(y))))$
 $t_2 = \text{suma}(\text{pred}(x), \text{suc}(y))$
 $t_3 = \text{pred}(\text{suma}(x, \text{suc}(y)))$

caso 3) la posición u_2 es prefijo de u_1 y u_2 corresponde a una posición de iz_1 en la que hay un símbolo de operación. Es decir, u_2 está en el "patrón" definido por iz_1 . Gráficamente corresponde a la situación:



Ejemplo: $R = \{ \text{pred}(\text{suc}(x)) \rightarrow x, \text{suma}(\text{pred}(x), y) \rightarrow \text{pred}(\text{suma}(x, y)) \}$
 $t = \text{suma}(\text{pred}(\text{suc}(x)), y)$
 $t_1 = \text{pred}(\text{suma}(\text{suc}(x), y))$
 $t_2 = \text{suma}(x, y)$

Ahora, al aplicar $iz_2 \rightarrow de_2$, se destruye el patrón de iz_1 con lo que la regla $iz_1 \rightarrow de_1$ no es, en principio, aplicable a t_2 . La confluencia en este caso depende de si las restantes reglas de R hacen o no converger los dos términos resultantes hacia un término común. Nótese que la causa del problema está en que hay dos reglas diferentes cuyos patrones no son disjuntos o se **superponen** (ver apartado 2).

Definición

Sean $iz_1 \rightarrow de_1, iz_2 \rightarrow de_2$ dos reglas de reescritura tales que $\text{var}(iz_1) \cap \text{var}(iz_2) = \emptyset$ e iz_2 se superpone a iz_1 en la posición $u \in \text{Pos}(iz_1)$ con sustitución σ . El par:

$$\langle \bar{\sigma}(de_1), \bar{\sigma}(iz_1)[u \leftarrow \bar{\sigma}(de_2)] \rangle$$

se denomina **par crítico** generado por las reglas $iz_1 \rightarrow de_1$ e $iz_2 \rightarrow de_2$.

La existencia de superposición entre dos reglas puede ser causa de no confluencia. El

término $\bar{\sigma}(iz_1)$, y cualquier término que contenga una instancia suya, puede reescribirse de dos modos distintos según se aplique una u otra regla. Cada componente del par crítico es precisamente el resultado de una de esas reescrituras.

Nótese que una regla puede superponerse a sí misma dando lugar a pares críticos. Véase el siguiente ejemplo:

$$\text{suma}(\text{suma}(x, y), z) \rightarrow \text{suma}(x, \text{suma}(y, z))$$

se superpone a sí misma en la posición 1 dando lugar, previo renombramiento de las variables, al término unificador:

$$\bar{\sigma}(iz) = \text{suma}(\text{suma}(\text{suma}(x', y'), z'), z)$$

usando la sustitución $\sigma = \{x \rightarrow \text{suma}(x', y'), y \rightarrow z'\}$. Reescribiendo $\bar{\sigma}(iz)$, se obtiene el par:

$$\langle \text{suma}(\text{suma}(x', y'), \text{suma}(z', z)), \text{suma}(\text{suma}(x', \text{suma}(y', z')), z) \rangle$$

Toda regla $iz \rightarrow de$ puede superponerse a sí misma en la posición λ dando lugar al par crítico trivial $\langle de, de \rangle$.

Diremos que un par crítico $\langle t_1, t_2 \rangle$ es convergente si existe un término t tal que $t_1 \rightarrow^*_R t$ y $t_2 \rightarrow^*_R t$. Cuando un conjunto de reglas da lugar, superponiéndolas entre sí, a pares críticos no triviales, el siguiente teorema nos dice que si todos estos pares críticos son convergentes, entonces el sistema de reescritura es localmente confluente.

Teorema de Knuth-Bendix

Un sistema de reescritura R es localmente confluente si y solo si todo par crítico $\langle t_1, t_2 \rangle$ es convergente.

Por lo tanto, si R es noetheriano, R es confluente si y solo si todo par crítico $\langle t_1, t_2 \rangle$ es convergente.

5. El algoritmo de completación

El teorema de Knuth-Bendix proporciona un procedimiento efectivo para decidir la confluencia de sistemas de reescritura noetherianos con un número finito de reglas. El procedimiento se resume en los siguientes pasos:

- calcular todas las posibles superposiciones de las reglas de R y crear todos los pares críticos no triviales.
- comprobar si todos los pares críticos convergen en el sistema R .
- R es confluente si y solo si todos convergen.

Ejemplo: Sea $R = \{$ 1) $\neg\neg x \rightarrow x$
2) $\neg(x \wedge y) \rightarrow \neg x \vee \neg y$
3) $\neg(x \vee y) \rightarrow \neg x \wedge \neg y$ $\}$

que es noetheriano. Da lugar a los siguientes pares críticos:

a) superponiendo (1) y (1) se obtiene $\neg\neg\neg x'$ que genera el par crítico $\langle \neg x', \neg x' \rangle$ trivialmente convergente.

b) superponiendo (1) y (2) se obtiene $\neg\neg(x' \wedge y)$ que genera $\langle x' \wedge y, \neg(\neg x' \vee \neg y) \rangle$. Aplicando (3) y (1) al segundo componente, éste se reescribe en la forma normal $x' \wedge y$, luego el par converge.

c) superponiendo (1) y (3) se obtiene $\neg\neg(x' \vee y)$ que genera $\langle x' \vee y, \neg(\neg x' \wedge \neg y) \rangle$. Aplicando (2) y (1) al segundo componente, se obtiene el término irreducible $x' \vee y$, luego el par converge.

Al no haber más superposiciones posibles, concluimos que el sistema es confluente y, por tanto, canónico. ♦

Si el sistema resultara ser no confluente, los pares críticos no convergentes nos dan un indicio de cual es la causa de la no confluencia. En efecto, por proceder de la reescritura de un término común, los dos componentes del par crítico son términos congruentes en la teoría ecuacional asociada al sistema de reescritura, es decir, si $\langle t_1, t_2 \rangle$ es un par crítico, $t_1 \leftrightarrow^*_R t_2$. Por tanto, añadir la regla $t_1 \rightarrow t_2$ o la $t_2 \rightarrow t_1$ no modifica la teoría definida por R y, en cambio, puede hacer confluente el sistema, ya que al menos ese par crítico se haría trivialmente convergente. Esta es la idea del algoritmo de completación. Antes de presentarlo, es necesario hacer algunas precisiones:

- para preservar la noetherianidad y, por tanto, no perder la equivalencia entre confluencia y confluencia local, las reglas añadidas $iz \rightarrow de$ han de estar orientadas de forma que $iz > de$ en el orden de reducción utilizado para demostrar la terminación finita del sistema.
- antes de añadirlo como regla, los dos componentes del par crítico han de estar reducidos a su forma normal. De este modo se acorta la duración del algoritmo y el número de reglas a añadir.
- las reglas añadidas pueden generar nuevas superposiciones y, por tanto, nuevos pares críticos cuya convergencia será necesario comprobar. Es decir, una sola iteración de cálculo de pares críticos no será en general suficiente. El algoritmo genera sucesivos sistemas de reescritura R, R', R'', \dots todos ellos noetherianos y definiendo la misma teoría ecuacional.
- los pares críticos no son otra cosa que ecuaciones válidas en la teoría asociada a R . Por tanto cabe tratar las ecuaciones E de una especificación como pares críticos y hacer que el algoritmo comience a trabajar con un conjunto vacío de reglas. Usado de este modo, el algoritmo es un "traductor" que convierte una especificación algebraica en un sistema de reescritura canónico.

Una versión simplificada del algoritmo podría ser la siguiente (E_0 es el conjunto inicial de ecuaciones):

```

E := E0 ; R := ∅ ; éxito := falso ; fallo := falso ;
mientras ¬éxito ∧ ¬fallo hacer
    caso E ≠ ∅ → orientar ecuaciones E obteniendo el conjunto de reglas RE
        si no es posible orientar alguna ecuación entonces fallo := cierto
        sino R := R ∪ RE fsi
    [] E = ∅ → calcular el conjunto PC de pares críticos no convergentes de R
        si PC = ∅ entonces éxito := cierto
        sino E := PC fsi
fcaso
fmientras

```

Si el algoritmo termina con éxito, el sistema final R es canónico. Si termina con fallo, no significa que E no pueda convertirse en un sistema canónico de reglas. Quizás fuera posible con otra elección del orden $>$.

Lamentablemente, el algoritmo puede no terminar, generando indefinidamente nuevos

pares críticos no convergentes y reglas que, a su vez, generan nuevos pares críticos, etc. En ese caso tampoco es posible afirmar que no exista un sistema canónico equivalente a E.

La corrección de la versión presentada, se deduce inmediatamente a partir de los resultados de los apartados previos. El invariante del bucle principal es que la teoría ecuacional definida por $E \cup R$ en curso es siempre la misma (y por tanto igual a la inicial E_0), y que R es noetheriano. Si el algoritmo termina con éxito, E es vacío y todos los pares críticos de R son convergentes lo que, unido al invariante, nos garantiza que R es canónico y define la misma teoría que E_0 .

Las versiones prácticas del algoritmo están notablemente mejoradas para aumentar la eficiencia. Las reglas se etiquetan con un entero, k : $iz_k \rightarrow de_k$, para describir el orden de aparición de las reglas en el sistema. Una de las mejoras es marcar las reglas para evitar recalcular pares críticos ya calculados. Se marca una regla cuando se han calculado todos sus pares críticos con las reglas que se han introducido previamente, esto es, las de etiqueta menor. Otra es convertir ecuaciones a reglas de forma incremental y calcular los pares críticos también de forma incremental. Cada vez que se añade una regla a R, se pueden simplificar las partes derechas o izquierdas de las reglas existentes. Si se simplifica la parte derecha de una regla es probable que se pierda la propiedad $iz > de$ por lo que habría que convertir la regla en ecuación. Si se simplifica la parte izquierda, la regla se conserva como tal y se consigue que las subsiguientes reescrituras sean más eficientes. Una nueva regla puede eliminar otras anteriores. Todas estas mejoras fueron propuestas en la versión de Gerard Huet (1981) que se presenta a continuación:

funcion completión (E: cjto_ecns; R: cjto_reglas; <: orden; n: entero) dev (cjto_ecns, cjto_reglas)

si $E \neq \emptyset$ entonces

seleccionar una ecuación $t_1=t_2$ de E

$fn1 := t_1 \downarrow$; $fn2 := t_2 \downarrow$;

si $fn1 = fn2$ entonces

(E, R) := completión (E- $\{t_1=t_2\}$, R, <, n)

sino

caso

$fn1 > fn2 \rightarrow iz := fn1$; $de := fn2$;

$fn2 > fn1 \rightarrow iz := fn2$; $de := fn1$;

otros \rightarrow parar con fallo

fcaso

(E, R) := simplificar (E, R, $iz \rightarrow de$);

(E, R) := completión (E := E - $\{t_1=t_2\}$; R := R \cup {n: $iz \rightarrow de$ }, <, n+1)

fsi

sino

si todas las reglas de R están marcadas entonces

parar con éxito

sino

seleccionar una regla $m: iz_m \rightarrow de_m$ no marcada de R;

E := pares_críticos (R, $m: iz_m \rightarrow de_m$);

marcar la regla $m: iz_m \rightarrow de_m$;

(E, R) := completión (E, R, <, n)

fsi

fsi

ffuncion dev (E, R)

funcion simplificar (E: cjto_ecns; R: cjto_reglas; $iz \rightarrow de$) dev (cjto_ecns; cjto_reglas)

L := {k | la regla de etiqueta k es reducible en su parte izquierda por $iz \rightarrow de$ };

E := E \cup { $iz'_k = de_k$ | k \in L y iz_k se reescribe en iz'_k por $iz \rightarrow de$ };

R := { $iz_k \rightarrow de'_k$ | k \notin L y de_k se reescribe en de'_k por $iz \rightarrow de$ }

ffuncion dev (E, R)

función pares_críticos (R: cjto_reglas; $m: iz_m \rightarrow de_m$) dev cjto_ecns;

calcula los pares críticos P de la regla $m: iz_m \rightarrow de_m$ con todas las reglas de etiqueta k < m

ffunción dev (P)

Dado un conjunto de ecuaciones E , se obtiene un sistema de reescritura equivalente R con la llamada "compleción $(E, \emptyset, \prec, 1)$ ".

Ejemplo: Sea el siguiente sistema ecuacional $E_0 = \{ (x.y).(y.z) = y \}$. Inicialmente el algoritmo orientaría esta ecuación de izquierda a derecha. La superposición de la única regla consigo misma en las posiciones 1 y 2 dan, en una primera iteración, los siguientes pares críticos no convergentes:

$$\langle y'.z', y'.((y'.z').z) \rangle \quad \text{y} \quad \langle x'.y', (x.(x'.y')).y' \rangle$$

que pueden ser orientados e incorporados a R , obteniéndose el sistema R' :

$$\{ \begin{array}{l} (x.y).(y.z) \rightarrow y \\ y'.((y'.z').z) \rightarrow y'.z' \\ (x.(x'.y')).y' \rightarrow x'.y' \end{array} \}$$

cuyos pares críticos son todos convergentes. Luego R' es canónico y equivalente a E_0 . ♦

Con ligeras modificaciones, el algoritmo de Knuth-Bendix puede servir también para demostrar *teoremas inductivos*. Un teorema inductivo es una ecuación $t_1 = t_2$ con variables, que sólo es válida en las álgebras generadas (entre ellas la inicial) que satisfacen la especificación SP. Sin embargo, algunas álgebras no generadas de Alg_{SP} pueden no satisfacerlo. Por tanto, no será deducible mediante el cálculo ecuacional ni, consiguientemente, mediante la reescritura, incluso disponiendo de un sistema canónico. Para demostrar un teorema inductivo, se añade éste a un sistema ya completado (canónico) y se ejecuta de nuevo el algoritmo de compleción. Si termina con éxito generando un nuevo sistema canónico en el que las reglas añadidas no introducen inconsistencias en la teoría ecuacional previa (por ejemplo, *cierto* \rightarrow *falso* sería una regla inconsistente con la teoría habitual de los booleanos), el teorema inductivo queda demostrado. Si termina con fallo o no termina, no es posible afirmar nada sobre el teorema.